\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: HW4

Author: F14076083 魏湧致 <zxc33991177@gmail.com>

Class: 資訊111 (乙班)

Description:

這題是要在float,double與其bit pattern互換，最主要的方法是要宣告一個整數指標來指向float，這樣才能進行位元運算的操作。將float或double換成bit pattern時要將或int或long long的1往左移並跟整數指標做 & 運算並print出bit pattern。將bit pattern變回double是將bit pattern各個左移並與long long指標做 | 運算將bit pattern放到double中最後print出結果，但變回float用同樣的方法輸出是0，所以改用union的方法，將float的sign,exponent,mantissa先算出來再與float union，才能print出正確結果。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Code:

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

typedef union //將m,e,s與float union

{

float f;

struct

{

unsigned int m : 23;

unsigned int e : 8;

unsigned int s : 1;

}r;

}flo;

int main(int argc, char \*argv[])

{

int t=atoi(argv[1]);

float num1=0;

double num2=0;

unsigned int\* pt1 = (int\*)&num1;

unsigned long long\* pt2 = (unsigned long long \*)&num2;

//將int與long long指標指向float與double

unsigned long long m=1;

if(t==1)

{

num1=atof(argv[2]);

for (int i=31; i>=0; i--) //迴圈內做

//(1<<i)&\*pt1的運算

{

if( (1<<i) & \*pt1 )

printf("1");

else

printf("0");

if(i==31 || i==23)

printf(" ");

}

printf("\n");

}

else if(t==2)

{

num2=atof(argv[2]);

for (int i=63; i>=0; i--)

{

if( (m<<i) & \*pt2 )

printf("1");

else

printf("0");

if(i==63 || i==52)

printf(" ");

}

printf("\n");

}

else if(t==3)

{

flo tra;

tra.r.s=argv[2]-'0'; //設定sign的值

tra.r.e=0;

tra.r.m=0;

for (int i=0; i<=7; i++) //設定exponent的值，

//將每個位數乘2的次方相加

{

tra.r.e = tra.r.e + (argv[3][i]-'0') \* pow(2,7-i);

}

for (int i=0; i<=22; i++) //設定mantissa的值

{

tra.r.m = tra.r.m + (argv[4][i]-'0') \* pow(2,22-i);

printf("",(argv[4][i]-'0') \* pow(2,22-i),tra.r.m);

}

printf("%f\n",tra.f);

}

else if(t==4)

{

int j=2;

int k=0;

for(int i=0; i <= 63; i++) //迴圈內將各位元左移

//並做 | 運算

{

\*pt2 = \*pt2 | (((unsigned long long)(argv[j][k++] - '0'))<<(63-i));

if(i==0 || i==11)

{

k=0;

j++;

}

}

printf("%f\n",num2);

}

return 0;

}Compilation:

gcc -lm -o hw4 hw4.c

Execution:

./hw4 1 85.125

./hw4 2 85.125

./hw4 3 0 10000101 01010100100000000000000

./hw4 4 0 10000000101 0101010010000000000000000000000000000000000000000000

Output:

0 10000101 01010100100000000000000

0 10000000101 0101010010000000000000000000000000000000000000000000

85.125000

85.125000

2

2-1 Yes, bit pattern為0 00000001 00000000000000000000000

2-2 0 00000000 00000000000000000000000

2-3 輸出的結果是相等，可能是因為設定的值超出可設定的位數所以是一

樣的數

2-4 NaN:0 11111111 01010100100000000000000

2-5 i. 3.141593

ii. 0.333333